

Lab 1. First Java programs

Welcome to your first lab for INFO1103 Introduction to Programming!

1. Introduction

Introduce yourself to the class. The tutor will start; when it is your turn, say who you are and something interesting about yourself that you want to share.

2. Login to SIT machines and creating directories

Log in to a PC in the lab and start up a terminal window. If there are problems with your login, your tutor should be able to help you.

Change the current directory to your U drive using the `cd` command. Make a new directory `info1103` for your lab work and then a subdirectory for this week's lab. It's best to be organised from the beginning!

```
> u:
> mkdir info1103
> cd info1103
> mkdir lab1
> cd lab1
```

3. HelloWorld

Open a text editor, e.g. Notepad++. Type the HelloWorld program exactly as below – the same indentation and spelling (remember that Java is case sensitive!). Save your file as `HelloWorld.java`.

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Go to the terminal window and compile your program:

```
> javac HelloWorld.java
```

If there are any errors, see if you can work out what they mean and fix them. If there are no errors, a class file is created (`HelloWorld.class`), you can check this by typing `dir`. The class file contains bytecode that is not very understandable for humans. You cannot edit this file but you can view it with:

```
> type HelloWorld.class
```

To run your program enter:

```
> java HelloWorld
```

Note: If you are not able to run your program and receive the following message: "Could not find or load main class HelloWorld", try this:

```
java -cp . HelloWorld
```

This command tells Java to look for the class file in the current directory, which is denoted with `."`. You can also update the `"CLASSPATH"` variable so that you don't need to use the `"java -cp"` version every time when you run Java from the command line. Type:

```
set
```

What is the value of the `CLASSPATH` variable? It should not include the current directory `."`. Now let's change this value so that it includes the current directory:

```
set CLASSPATH=.;%CLASSPATH%
```

And let's check that it is done:

```
set
```

Now let's verify that we don't need to use the `"-cp"` option:

```
java HelloWorld
```

It should run without any problems.

You can also set the `CLASSPATH` variable in another way, using the `"Properties"` option in `"My Computer"` for Windows computers. For more information, please see:

<https://docs.oracle.com/javase/tutorial/essential/environment/paths.html>

Back to our `HelloWorld` program. Introduce some syntax errors. See the lecture slides if you don't remember what syntax errors are. Compile again. What happens?

4. Greeting in another language

Write a program called `Greeting.java` that prints a greeting in another language of your choice. For example: "Italian greeting: Ciao! Come stai?"

5. Name in box

Write a program called `NameInABox.java` that prints your name in a box like this:

```
+-----+
| David |
+-----+
```

6. Face

Write a program called `Face.java` that prints a face similar to this one (you can make your own):

```

      /////
     .'.  '.
    | 0  0 |
    |  <  |
    |  _  |
     '.  .'
    -...-

```

Note: if you want to print a backslash `\` you need to put two backslashes `\\` in a row.

7. HelloWorld in Eclipse

Start Eclipse on your computer. We use projects and packages to organise our Java programs. Let's keep all the labs in one project, and in this project we will have one package for each week's lab. To do this, firstly create a Java project called `Labs`. Make sure there is a source folder: it usually appears by default and should be called `src`. Secondly, in the `Labs` Java project, create a package called `lab1`. Next, make a new class file in that package called `HelloWorld.java`. Copy your HelloWorld code in to that file from the file where you saved it earlier. If you need help with creating projects and packages, see the "INFO1103 User Manual" – it is more detailed and contains useful screenshots.

Are there any errors? In Eclipse, syntax errors and other compiler errors are highlighted. You can move the mouse over the error image or the warning image and a message should pop up to say what the problem is. Once your code is error-free, run the HelloWorld program from within Eclipse. To do that, make sure your `HelloWorld.java` file is selected on the left, then press the run button.

Introduce a syntax error into your program, e.g. a missing `“;”`. Eclipse will alert you almost immediately where there is a syntax error, or where the compilation would fail.

8. Improved HelloWorld in Eclipse

Run `FirstProgram.java` in Eclipse, the improved version of HelloWorld from the lecture. Create a new class `FirstProgram` in the package `lab1`. Type the code from the lecture slides and pay attention to the indentation, spelling and curly braces.

Modify the program so that it adds three numbers instead of two. Compile the program so that you receive no compiler error messages. Then run your program.

9. To do at home

Install Java and Eclipse on your computer at home and learn how to use Eclipse. The "INFO1103 User Manual" on eLearning shows you how to do this.